



ARCH 4342

ASSIGNMENT 3

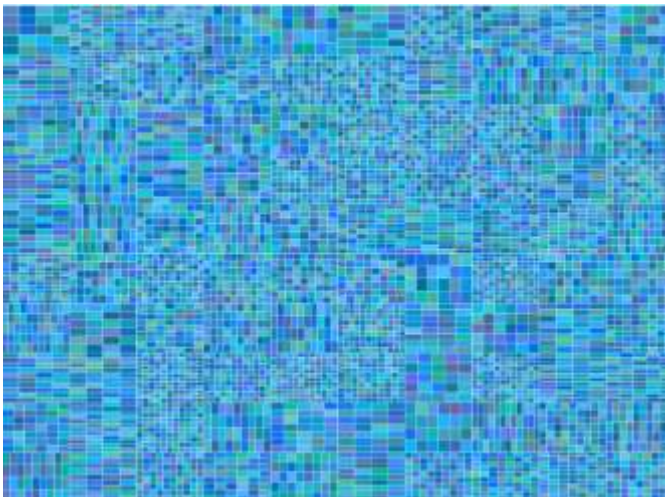
Υπολογιστικές Προσεγγίσεις στις
Δημιουργικές Τέχνες και Επιστήμες

Κατερίνα Σκρέτα
ar24742

Πρόβλημα 1 / Arrays & Iteration

Ψευδοκώδικας

1. Όρισε μεταβλητές cols, rows
2. Στο setup δημιούργησε καμβά στο μέγεθος του παραθύρου και θέσε λευκό φόντο
3. Υπολόγισε cell_width και cell_height Για κάθε κελί του βασικού πλέγματος (δύο εμφωλευμένοι βρόχοι με i και j):
4. Σχεδίασε το βασικό κελί
5. Επίλεξε τυχαίο αριθμό εσωτερικών στηλών
6. Επίλεξε τυχαίο αριθμό εσωτερικών γραμμών
7. Υπολόγισε το υποκελί
8. Όρισε περίγραμμα και χρώμα κελιών



```
1 var cols = 10;
2 var rows = 10;
3
4 function setup() {
5   createCanvas(windowWidth, windowHeight);
6   background(255);
7   // υπολογισμός πλάτους και ύψους κάθε κελιού του βασικού πλέγματος
8   cell_width = width / cols;
9   cell_height = height / rows;
10
11   //βρόχοι για κάθε κελί του βασικού πλέγματος
12   for(var i=0; i<cols; i++){
13     for(var j=0; j<rows; j++){
14       rect(i * cell_width, j * cell_height, cell_width, cell_height);
15       // βασικό κελί
16
17       // δημιουργία τυχαίων μικρότερων γραμμών και στηλών μέσα στο κελί
18       var innerCols = int(random(4,13));
19       var innerRows = int(random(4,13));
20       var innerCellWidth = cell_width / innerCols;
21       var innerCellHeight = cell_height / innerRows;
22
23       for(var m=0; m<innerCols; m++){
24         for(var n=0; n<innerRows; n++){
25
26           let x = i * cell_width + m * innerCellWidth;
27           let y = j * cell_height + n * innerCellHeight;
28
29           strokeWeight(0.3);
30           stroke(255);
31           fill(random(0,100), random(100,200), random(350,255));
32           rect(x, y, innerCellWidth, innerCellHeight);
33           //Εχρησιμοποιώ υποκελί
34
35
36
37
38
39
40
41 function draw() {
42 }
43
44 function mousePressed() {
45 }
46
47 }
```

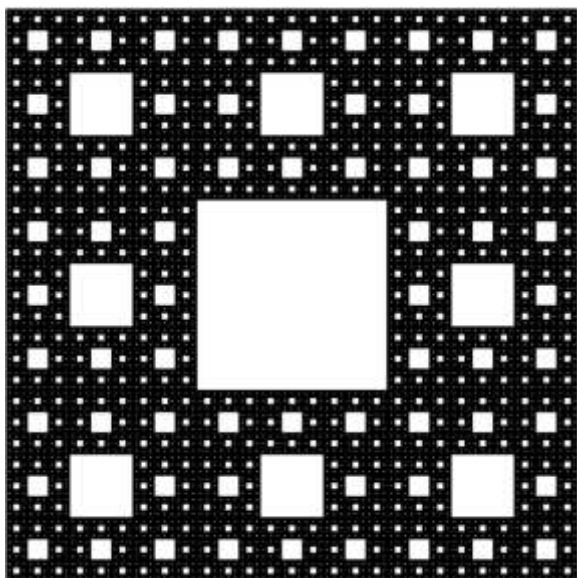
<https://editor.p5js.org/askreta/full/VEGsiC-P5>

<https://editor.p5js.org/askreta/sketches/VEGs>

Πρόβλημα 2 / Recursion: Fractal City

Ψευδοκώδικας

1. Δημιούργησε καμβά διαστάσεων 500 x 500.
2. Θέσε το φόντο του καμβά σε γκρι χρώμα
4. Σχεδίασε ένα μαύρο τετράγωνο στο κέντρο του καμβά και όρισε το χρώμα του μαύρο.
5. Κάλεσε τη συνάρτηση `fractal(x,y,s)`
6. Όρισε το χρώμα της λευκό.
7. Αν $s < 3$ τερμάτισε την συνάρτηση.
8. Αν δεν ισχύει:
9. υπολόγισε το μέγεθος κάθε μικρού τετραγώνου
10. Σχεδίασε ένα λευκό τετράγωνο στο κέντρο του ήδη υπάρχοντος τετραγώνου
11. Επανέλαβε για κάθε θέση από 0-2
12. Αν η θέση είναι στο κέντρο του καμβά παραλείπω το τετράγωνο αυτό
13. Αλλιώς ξανά καλώ την `fractal` για το νέο υποτετράγωνο



```
1* function setup() {  
2   createCanvas(500, 500);  
3   background(220);  
4   fill(0);  
5   rect(0,0,500);  
6   fill(255);  
7   fractal(0,0,500);  
8 }  
9  
10  
11* function draw() {  
12 }  
13  
14* function fractal(x,y,s){  
15  
16   if (s<3) return; // χωρίς αγωγή γιατί περιέχει μια εντολή  
17  
18   let newrect= s/3  
19   rect(x + newrect,y + newrect, newrect, newrect);  
20  
21*   for (let i=0; i < 3; i++){  
22*     for (let j=0; j < 3; j++){  
23*       if ((i===1 && j===1) continue; // Χρησιμοποιώ την εντολή αυτή για να παραλείψω  
24*       το κεντρικό rect από την διαδικασία σχεδίασης της fractal. Το i===1 και j===1 στο  
25*       πλέγμα 3x3 αντιστοιχεί στο μεσαίο rect όπου ξεκινάει με i=0 και j=0, το οποίο θ  
26*       fractal(x + i * newrect, y + j * newrect, newrect);  
27*     }  
28*   }  
29 }  
30 }
```

https://editor.p5js.org/askreta/full/0E8y_Vf0T

https://editor.p5js.org/askreta/sketches/0E8y_Vf0T

ΜΕΡΟΣ ΤΡΙΤΟ

Computer Game

Ψευδοκώδικας

1. Όρισε τις μεταβλητές για τη θέση του φιδιού, την κατεύθυνση, το μήκος σώματος, τα τρόφιμα και τα κτίρια.
2. Δημιούργησε καμβά πλήρους παραθύρου και όρισε το frame rate.
3. Τοποθέτησε το φίδι στο κέντρο και πρόσθεσέ το στη συλλογή snakeBody.
4. Δημιούργησε τυχαίες θέσεις για τρόφιμα και αποθήκευσέ τες σε συλλογή.
5. Δημιούργησε κτίρια με τυχαία θέση και μέγεθος και αποθήκευσέ τα σε συλλογή.
6. Καθάρισε το φόντο με μαύρο.
7. Σχεδίασε όλα τα κτίρια από τη λίστα.
8. Σχεδίασε όλα τα τρόφιμα ως μικρά τετράγωνα.
9. Υπολόγισε την επόμενη θέση του φιδιού σύμφωνα με την κατεύθυνση.
10. Πρόσθεσε τη τρέχουσα θέση στην αρχή της λίστας σώματος.
11. Αν το σώμα ξεπερνά το μέγιστο μήκος, αφαίρεσε το τελευταίο τμήμα.
12. Αν το φίδι αγγίζει κάποιο τρόφιμο, αφαίρεσέ το και αύξησε το μήκος του σώματός του.
13. Σχεδίασε κάθε τμήμα του σώματος του φιδιού.
14. Αν η επόμενη θέση είναι εκτός οθόνης, επανάφερε το φίδι στην αντίθετη πλευρά.
15. Αν η επόμενη θέση είναι μέσα σε κτίριο, μην ενημερώνεις τη θέση και σταμάτα.
16. - Διαφορετικά, ενημέρωσε τη θέση του φιδιού με τις νέες συντεταγμένες.
17. Άλλαξε την κατεύθυνση του φιδιού σύμφωνα με το πληκτρολόγιο.
18. Στην createBuildings σχεδίασε και αποθήκευσε κάθε κτίριο σε συλλογή.
19. Στην drawBuildings, σχεδίασε κάθε κτίριο ως λευκό ορθογώνιο.
20. Στην insideBuilding, έλεγξε αν οι συντεταγμένες βρίσκονται μέσα σε κάποιο κτίριο και επέστρεψε true ή false.

<https://editor.p5js.org/askreta/full/4NDjqG1Wr>
<https://editor.p5js.org/askreta/sketches/4NDjqG1Wr>

```

1 //Ορίζω variables
2 let x1;
3 let y1;
4 let step = 10;
5 let fwd_x = 0;
6 let fwd_y = 0;
7
8 let snakeBody = []; // μια λίστα από πίνακες με κάθε τμήμα του φιδιού
9 let bodyLength = 0;
10
11 let foodItems = [];
12 let totalFood = 100;
13
14 let buildingOutlines = [];
15 let totalBuildings = 0;
16
17 function setup() {
18   createCanvas(windowWidth, windowHeight);
19
20   frameRate(10); // θέτω frames ανά δευτερόλεπτο για να ελεγχώ την ταχύτητα του φιδιού
21
22   // Θέτω τις αρχικές συντεταγμένες του φιδιού στο κέντρο της οθόνης
23   x1 = width/2;
24   y1 = height/2;
25
26   snakeBody.push([x1, y1]);
27
28   // Δημιουργώ τυχαίες θέσεις στον χώρο για τα τρόφιμα
29   for (let i=0; i<totalFood; i++){
30     let x2 = random(width);
31     let y2 = random(height);
32     foodItems.push([x2, y2]);
33   }
34
35   createBuildings();
36
37 }
38
39 function draw() {
40   background(0);
41
42   drawBuildings();
43
44   // Ξεκινάω να τρώω
45   for (let j=0; j<foodItems.length; j++){
46     let food = foodItems[j];
47     let x3 = food[0];
48     let y3 = food[1];
49     fill(255);
50     stroke(255, 0, 0);
51     rect(x3, y3, 7, 7);
52   }
53
54   //Υπολογίζω την τρέχουσα θέση του φιδιού για να με βοηθήσει έπειτα όταν να μην αναγκαστεί το φίδι μέσο στα κτίρια
55   let nextX = x1 + fwd_x;
56   let nextY = y1 + fwd_y;
57
58   snakeBody.splice(0, 0, [x1, y1]); // η λίστα προσδίδει νέο τμήμα στην αρχή
59   if (snakeBody.length > bodyLength) {
60     snakeBody.pop();
61   }
62
63   // Ελέγγω εάν έχει καταναλωθεί τροφή
64   for (let n = foodItems.length - 1; n >= 0; n--) {
65     let food = foodItems[n];
66     let x4 = food[0];
67     let y4 = food[1];
68
69     let d = dist(x1, y1, x4, y4);
70     if (d == step) {
71       foodItems.splice(n, 1); // Αφαιρώ την τροφή που έφαγε
72       bodyLength -= 1; // Αυξάνω το μέγεθος του φιδιού
73     }
74   }
75
76   // Ξεκινάω το σώμα του
77   fill(255, 0, 0);
78
79   for (let k=0; k < snakeBody.length; k++){
80     let segment = snakeBody[k];
81     let x5 = segment[0];
82     let y5 = segment[1];
83
84     rect(x5, y5, step, step);
85   }
86
87   // Σε περίπτωση που βγει εκτός οθόνης μπαίνει ξανά μέσα
88   if (nextX > width) {
89     nextX = 0;
90   } else if (nextX < 0) {
91     nextX = width;
92   }
93
94   if (nextY > height) {
95     nextY = 0;
96   } else if (nextY < 0) {
97     nextY = height;
98   }
99
100   // Αν το επόμενο βήμα που υπολογίστηκε παραπάνω βρίσκεται μέσα στο κτίριο τότε το φίδι δεν μπορεί να προχωρήσει ομαλά εκεί
101   if (isInBuilding(nextX, nextY)) {
102     return;
103   }
104
105   x1 = nextX;
106   y1 = nextY;
107 }
108
109 function keyPressed() {
110   // Ελέγχος πορς κίνησης
111   if (keyCode === UP_ARROW) {
112     fwd_x = 0;
113     fwd_y = -step;
114   } else if (keyCode === DOWN_ARROW) {
115     fwd_x = 0;
116     fwd_y = step;
117   } else if (keyCode === LEFT_ARROW) {
118     fwd_x = -step;
119     fwd_y = 0;
120   } else if (keyCode === RIGHT_ARROW) {
121     fwd_x = step;
122     fwd_y = 0;
123   }

```

```

123
124 }
125
126 * function createBuildings() {
127   // Δημιουργία κτιρίων σε random θέσεις και με random διαστάσεις
128   * for(let m = 0; m < totalBuildings; m++) {
129     let x0 = random(width);
130     let y0 = random(height);
131     let wiidth = random(50,90);
132     let heeheight = random(50,90);
133
134     buildingOutlines.push([x0, y0, wiidth, heeheight]);
135     let b = buildingOutlines[m];
136     let x = b[0];
137     let y = b[1];
138     let z = b[2];
139     let f = b[3];
140   }
141 }
142
143 * function drawBuildings() {
144   // Σχεδιασμός κτιρίων
145   fill(250);
146   noStroke();
147
148   * for (let l = 0; l < buildingOutlines.length; l++) {
149     let b = buildingOutlines[l];
150     rect(b[0], b[1], b[2], b[3]);
151   }
152 }
153
154 // Έλεγχος θέσης-είναι μέσα στο κτίριο ή δεν είναι
155 * function insideBuilding(x, y) {
156   * for (let h = 0; h < buildingOutlines.length; h++) {
157     let b = buildingOutlines[h];
158     if (x >= b[0] && x < b[0] + b[2] &&
159         y >= b[1] && y < b[1] + b[3]) {
160       return true; //είναι
161     }
162   }
163   return false;
164 }

```

